

Amendments to the Specification

Please replace the paragraph at page 5, lines 3-16 with the following:

Further, as the amount of image data to be processed increases and the data processing method becomes complicated, the amount of data to be processed at once increases, and the number of processor elements needed increases. When the number of processor elements increases, the number of test patterns needed for logical testing, IC testing and so ~~fourth~~ forth increases. In order to perform a test performed on a single processor element on all the processor elements, the number of test patterns the same as the number of the processor elements are needed. Further, it is necessary to provide circuits needed for testing and ports through which test results are output for all the processor elements.

Please replace the paragraph at page 7, lines 5-13 with the following:

In a case where such table conversion is employed in an SIMD processor, the table is needed for each unit of operation. For example, when the SIMD processor includes 256 processor elements (PEs) and performs table conversion of 8 bits, a table RAM of 256 bytes is needed for each operation unit. Accordingly, a total of 256 table RAMs are needed. Accordingly, the SIMD processor is very expensive. In order to solve this problem, various methods have been proposed.

Please replace the paragraph bridging pages 10 and 11 with the following:

wherein the global processor provides an ~~instructions~~ instruction for setting processor-element numbers to all the processor elements, outputs a control signal through

execution of the instructions, and uses the processor-element number corresponding to each processor element as an input value of the operation array.

Please replace the paragraph at page 14, lines 12-22 with the following:

In the related art, when ~~data~~ data transfer for a certain range of processor elements is rendered, execution condition flags for the range are, set one by one, and, according to the set execution condition flags, whether or not transfer is to be made is determined. Accordingly, the number of cycles for the number of times of transfer is needed. However, when employing the above-mentioned configuration according to the present invention, by using MGAA instruction by which processor elements in a certain range can be specified, the same processing can be rendered through one cycle.

Please replace the paragraph bridging pages 14 and 15 with the following:

Thus, by using the MGAA instruction, the comparator shown in FIG. 5 is provided for each processor element, the comparator compares the PE number held by each processor element with the upper ~~value~~ value and lower value specified through the immediate-value operands. Then, when it is within the range, the operation of the processor element is performed. Execution/non-execution is controlled without the A-register updating the result of the ALU (without updating the latch signal).

Please replace the paragraph at page 15, lines 8-20 with the following:

The data transfer may be rendered through specifying bits such as to specify processor elements matching processor-element numbers expressed in binary notation; and specifying

processor elements by masking arbitrary bits of the thus-specified bits, through specifying the immediate values by using the operands. For this purpose, a pattern matching block is provided for selecting processor elements by using the given ~~processor-element~~ processor-element number, specifying a range using a bit pattern of immediate values, and specifying bit-masking through masking by using a bit pattern of immediate values. Thereby, specific processor elements are controlled.

Please replace the paragraph at page 17, lines 6-15 with the following:

Each processor element may comprise a plurality of flag bits for controlling as to whether or not the result of the operation is to be stored in a register, which flag bits can be set/reset according to the result of the operation or a control signal from the global processor. A logical operation is performed on the state of ~~flag~~ flag bits before the setting/resetting and the value to be newly set/reset. An AND/OR logical operator is provided, and the value obtained through the operation is set:/reset into the flag bits.

Please replace the paragraph at page 19, lines 12-13 with the following:

A parallel processor according to another ~~aspect~~ aspect of the present invention comprises:

Please replace the paragraph at page 19, lines 22-25 with the following:

data from a table ~~memory~~ memory is stored in at least one register of each of a plurality of processor elements having the same contents of the operation-result flag, simultaneously.

Please replace the paragraph at page 20, lines 7-10 with the following:

data after conversion from the table ~~memroy~~ memory may be stored in the at least one register of each of the plurality of processor elements having the same contents of the operation-result flag, simultaneously.

Please replace the paragraph at page 20, lines 17-21 with the following:

the data after conversion from the table ~~memroy~~ memory may be stored in the at least one register of each of the plurality of processor elements having the same contents of the operation-result flag, simultaneously.

Please replace the paragraph bridging pages 20 and 21 with the following:

In the configuration, a data transfer bus connecting the table ~~memroy~~ memory to the register of the register file of each processor element, and a control part controlling data transfer from the data transfer bus to the register may be provided; and

Please replace the paragraph at page 21, lines 16-20 with the following:

Data of a table memory to be stored in a plurality of registers may be stored in a memory built in the global processor, and the ~~memroy~~ memory can also be used as a memory for storing data processed in operation of the global processor.

Please replace the paragraph at page 21, lines 21-22 with the following:

An image processing apparatus ~~aeccordintg~~ according to the present invention, comprises:

Please replace the paragraph at page 22, lines 12-16 with the following:

data after conversion for non-linear processing from a table ~~memroy~~ memory is stored in at least one register of each of a plurality of processor elements having the same contents of the operation-result flag, simultaneously; and

Please replace the paragraph bridging pages 22 and 23 with the following:

the data after conversion from the table ~~memroy~~ memory may be stored in the at least one register of each of the plurality of processor elements having the same contents of the operation-result flag, simultaneously.

Please replace the paragraph at page 23, lines 5-9 with the following:

A data transfer bus connecting the table ~~memroy~~ memory to the register of the register file of each processor element, and a control part controlling data transfer from the data transfer bus to the register may be provided; and

Please replace the paragraph at page 27, lines 14-23 with the following:

As shown in FIG. 1, the SIMD processor 1 includes a global processor (GP) 2, a processor-element block 3 including 256 processor elements (PEs) 3a (FIG. 2), and an interface 4. The interface 4 gives data to undergo operation processing provided from an external image scanner, for example, to an input/output register file 31, and, also, transfers

data having undergone the operation processing to an external printer or the like from the register file 31, based on instructions given by the global processor 2.

Please replace the paragraph at page 30, lines 8-16 with the following:

The SCU 22 of the global processor 2 sends to the interface 4 operation setting data, commands and so forth for data transfer. The interface 4, based on the operation setting data and commands sent from the SCU 22, generates an address control signal for specifying an address for the processor elements 3a, a read/write control signal for giving instructions of data read/write to registers 31b (FIG. 3) of the processor elements 3a and a clock control signal for providing a clock signal.

Please replace the paragraph at page 30, lines 17-25 with the following:

The read/write control signal includes a write control signal for obtaining data to undergo operation processing from a data bus and causing the data to be held by the register files 31 of the processor elements 3a. The read/write control signal also includes a read control signal for giving instructions to the ~~resisters~~ registers such that data having undergone the operation processing and held by the register files 31 of the processor elements 3a to be given to the data bus.

Please replace the paragraph at page 38, lines 18-21 with the following:

The operation array 36 of each processor element 3a includes the ~~multiplexer~~ multiplexer 32, shift and expansion circuit 33, 16-bit ALU 34, 16-bit A-register 35a and 16-bit F-register 35b.

Please replace the paragraph bridging pages 44 and 45 with the following:

When the above-mentioned PE shift is rendered, in the related art, in an SIMD processor, only same data can be set in all processor elements. Accordingly, it is necessary to transfer data to the A-register for each PE, i.e., PE0, PE1, PE2, . . . , PE by PE, so as to set different values there. If the same ~~value~~ value in all the processor elements results from rendering the PE shift, it is not possible to determine which PE the data has been transferred from. However, in the first embodiment of the present invention, by using the LDPN instruction, it is possible to set different PE numbers in the A-registers of the processor elements at once. By using the LDPN instruction, values of respective PE numbers are set in the A-registers 35a of all the processor elements 3a, for ~~example~~ example. In this example, 0, 1, 2, 3, . . . are set in the A-registers in the ~~ascending~~ ascending order of the processor elements. Then, data is stored in any register of the register file of each processor element by the PE shift. For ~~example~~ example, when the PE shift of lowering by two is rendered and the data is stored in the register R1 of each processor element, 2, 3, 4, 5, . . . are then stored in the R0-registers in the ~~ascending~~ ascending order of the processor elements. At last, when the A-register is ~~compare~~ compared with the R0-register in data stored therein,

$$(\text{value in A-register}) - (\text{value in R0-register}) = 2$$

is obtained in each processor element. A value other than 2 is obtained in a processor element in which the PE shift is not rendered properly.

Please replace the paragraph bridging pages 45 and 46 with the following:

The LDPN instruction may be used for operation specifying every n-th processor element 3a. For example, when every fifth processor element PE0, PE5, PE10, is attempted

to be selected, PE numbers are set in all the processor elements 3a by using the LDPN instruction. When the values of A-registers 35a of PE0, PE1, PE2, . . . , PE255 are set in order, these values are 0, 1, 2, . . . , 255. Then, the value of each A-register 35a is divided by 5 and the remainder thereof is stored (This operation can be rendered by repeating subtracting operation by the ALU, and the value finally remaining in the A-register 35a is the remainder. Thus, a removing method or a retracting method of dividing operation is employed therefor.). Thereby, the values become 0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1, . . . , 0. Then, CMP instruction (~~comparison~~ comparison between the data in the A-register 35a and data in the register file 31 is made, and the result thereof is reflected in the specific bit of the T-register 35c) of the PE instruction is used. Thereby, it is possible to select the subsequent processor elements 3a of operation.

Please replace the paragraph bridging pages 46 and 47 with the following:

By the MGAA instruction, the value of the G0-register is transferred to the A-registers 35a of the processor elements PE_i through PE_j ($i \leq j$; $i, j = 0, 1, \dots, 255$). Addressing may be immediate addressing or register addressing, and, for example, is ~~described~~ described as follows:

Please replace the paragraph bridging pages 58 and 59 with the following:

In the first embodiment shown in FIG. 3, through single addressing, data can be transferred to the processor elements 3a assigned even numbers of the SIMD processor 1 from an external memory, and, also, data transfer can be made to the processor elements 3a assigned odd numbers. However, a method of inputting/outputting data to/from the SIMD

processor 1 externally is not limited to this method. The present invention may also be applied to an SIMD processor 1 such that, for ~~example~~ example, as shown in FIG. 6, without distinguishing the processor elements 3a of the SIMD processor 1 between those assigned even numbers and those assigned odd numbers, data is transferred in sequence by addressing. That is, as shown in FIG. 6 (showing a first variant embodiment of the above-described first embodiment), register controllers 31a are connected with the interface 4 via an address bus 41a, a read/write-signal line 45c, and a clock-signal line 41c. Each of the register controllers 31a, when receiving an addressing signal from the interface 4 via the address bus 41a, decodes the addressing signal. Then, when the address obtained from decoding coincides with the address assigned to itself, receives a read/write instruction signal via the read/write-signal line 45c in synchronization with a clock signal given via the clock-signal line 41c. The read/write signal is also given to a register 31b.

Please replace the paragraph bridging pages 72 and 73 with the following:

The operation array 136 includes the ~~multiplexer~~ multiplexer 132, shift and expansion circuit 133, 16-bit ALU 134, 16-bit A-register 135a and 16-bit F-register 135b.

Please replace the paragraph at page 75, lines 10-14 with the following:

Further, by the condition register (T) not shown in the figure, ~~valid/invalid~~ of execution of operation is controlled for each processor element 103a, and, thereby, it is possible to select specific processor elements (PE) 103a as targets of the operation.

Please replace the paragraph bridging pages 85 and 86 with the following:

(2) The global processor 102 inputs data to be converted, to the ALU 134 via the immediate data bus 141f. To the other input terminal of the ALU 134, the data stored in the A-register 135a in (1) is input. Then, the ALU 134 performs value comparison operation processing thereon. The comparison result is stored in an arbitrary bit of the condition register 135c of 8 bits as condition satisfaction when the data to undergo the operation processing is larger. Further, for the PE 103a for which the condition is satisfied, the ~~value~~ value obtained from subtracting the data to be converted from the data in the A-register 135a is temporarily stored in the register (R1) as differential data from the data to be converted.

Please replace the paragraph at page 97, lines 10-24 with the following:

FIG. 19 shows one example of the FIFO 107, and FIG. 20 shows one ~~example~~ example of the FIFO 108. The FIFOs 107 and 108 include ~~memory~~ memory controllers 172, 182 and buffer memories 171, 181. Input of external data is rendered in a manner such that data is stored in the buffer memory 171 via the memory controller 172, and, when image data for one scan line is stored in the buffer memory 171, the memory ~~controller~~ controller 172 transfers the data to the ~~processor-element~~ processor-element block 103 therefrom. External data output is rendered in a manner such that data is stored into the buffer ~~memory~~ memory 181 via the memory ~~controller~~ controller 182 from the processor-element block 103. When data for one scan line is stored in the buffer memory 181, the memory controller 182 transfers the data externally therefrom.